

Japanese Patent Laid-open No. 2001-1230 A

Publication date : January 9, 2001

Applicant : Toyoda Machine Works, Ltd.

Title : MACHINING PROGRAM CREATION METHOD FOR TWO-SPINDLE

5 MACHINE TOOL

[0023] The present invention provides a method for creating a NC program for both machining units LM and RM, when the work W mounted on a single table 41 is machined by coordinating two spindle heads 31 and 32. To execute this method, the CPU 51 is provided with, as shown in Fig. 3, software for executing coordination information adding processing 64. The program creation method according to the present invention is performed according to a procedure described below, by changing an operation mode of the CNC unit 50 to a program creation mode.

15 [0024] As shown in Fig. 4, as a first stage, a first and a second NC programs NP1 and NP2 for machining the work W are respectively created by individually operating the left and right machining units LM and RM. As the creation order, either the program NP1 or the program NP2 can be created first. Fig. 5 is an example of the both programs NP1 and NP2 for drilling and tapping, in association with machining positions a to g of the work W. In the machining example, the program NP1 for the left machining unit LM is formed of eight data blocks with sequence numbers N10 to N80, and programmed so as to drill prepared holes in the process of returning from a machining position d closest to the right machining unit RM to a starting point thereof in order of d → g → c → f Likewise, the program NP2 for the right machining unit RM is formed of eight data blocks with sequence numbers N10 to N80, and programmed so as to perform tapping with respect to the prepared holes in the process of moving toward the opponent, that is, the left machining unit LM, in the same order as the machining sequence of drilling.

30 [0025] The both programs NP1 and NP2 are created in an interactive manner such that the operator talks with the CPU 51, for example, by inputting an input

item instructed on a display screen 543 by using the ten keys 542, under execution of the input/output support processing 61. The respective NC programs NP1 and NP2 created in this manner are stored in the memory 52 for each one data block in a well-known EIA code format. For example, commands and data are input to each data block subsequent to a sequence number, in order of a control command (a so-called G code command), spindle-specifying data, target position data, a speed command, an M function command, and a block termination command. The respective NC programs NP1 and NP2 can be created by using a code format other than the EIA code.

[0026] Fig. 6 depicts a program example when the left and right machining units LM and RM take partial charge of a plurality of hole drilling operations. In this case, the NP1 for the left machining unit LM is programmed so as to perform hole drilling sequentially to a group of machining positions shown on the upper side of Fig. 6, in the process of returning from the machining position d closest to the right machining unit RM to the starting point thereof in order of $d \rightarrow c \rightarrow b \rightarrow a$. The program NP2 for the right machining unit RM is programmed so as to perform hole drilling sequentially to a group of machining positions shown on the lower side of Fig. 6, in the process of moving toward the opponent, that is, the left machining unit LM. In the respective NC programs shown in Figs. 5 and 6, a rotation speed command of the tool spindle is given to the first block. A tool replacement command is given to the previous stage of the first block of the respective NC programs, but these are omitted in Figs. 5 and 6. Furthermore, in Figs. 5 and 6, in the stage in which the programs NP1 and NP2 for single operation are created, a machining permission command MPC and a machining permission wait command PWC have not been inserted yet.

[0027] As shown in Fig. 4, in the machining program creation method of the present invention, debugging of the respective NC programs NP1 and NP2 is executed as a second stage. An object of the debugging processing DBG1 is to remove a problem (bug) included in the respective NC programs NP1 and NP2 when the respective NC programs are executed separately by the CNC unit 50. This processing can be executed by software simulation for checking program operation by displaying a track of the tool moving with respect to the work W on

the display screen 543 based on the respective NC programs NP1 and NP2, without actually operating the machine tool 10. However, when the debugging is to be performed more reliably, the respective NC programs NP1 and NP2 are actually executed by the CNC unit 50 separately, to execute checking by
5 moving the tool with respect to the work W. In this case, a no-load running test in which the tool is not brought into contact with the work W can be used, however, when trial machining in which the work W is machined by the tool is performed, the dimension of the work W, which has been subjected to the trial machining, is measured, to determine the quality of the NC programs NP1 and
10 NP2.

[0028] Subsequent to the debugging DBG1 for executing the respective NC programs NP1 and NP2 individually, in the third stage, coordination information is added to the NC programs NP1 and NP2 for operating the both machining units LM and RM simultaneously in coordination with each other. This
15 processing is explained with reference to Figs. 5 and 6. In the case of Fig. 5 in which drilling and tapping are assigned to the both machining units LM and RM, a plurality of machining permission commands MPC individually identified are added to the first NC programs NP1, which precedes the machining operation (hereinafter occasionally referred to as preceding machining program), and a
20 plurality of machining permission wait commands PWC individually identified are added to the second NC programs NP2, which starts the machining operation tardily (hereinafter occasionally referred to as subsequent machining program). Addition of the machining permission commands MPC and the machining permission wait commands PWC is carried out in the following
25 manner. That is, the data blocks in the subsequent machining program NP2 is sequentially selected as a comparison reference, the comparison reference block is compared with a data block to be compared of the preceding machining program NP1, which is sequentially selected, and the machining permission commands MPC and the machining permission wait commands PWC are
30 added based on the comparison result.

[0029] More specifically, the top data block "N10 d machining" of the subsequent machining program NP2 is selected as the comparison reference

block. Furthermore, as the data block to be compared, the first data block "N10 d machining", the second data block "N20 g machining", the third data block "N30 c machining", and the fourth data block "N40 f machining", of the preceding machining program NP1, are sequentially selected. When an interval between target positions on an X axis of the selected both blocks is equal to or smaller than a predetermined value K for avoiding interference, it is determined that there is an interference, and when the interval is equal to or larger than the predetermined value K, it is determined that there is no interference. The predetermined value K is preset to a value, at which even if a distance in the left and right direction between the spindles 29 and 40 of the both spindle heads 31 and 32 reaches this value, there is a slight space between the opposite side faces of the both spindle heads 31 and 32. As for the selected block of the preceding machining program NP1, if the block is near the top block, it is determined that there is interference. However, in the process of shifting the selected block toward the last block side, a block determined to have no interference is extracted. When this determination changes, a machining permission command MPC-d for permitting execution of the top block of the subsequent machining program NP2 is inserted into the previous stage of the block "N50 b machining", for which determination changes to "no interference". Furthermore, a machining permission wait command PWC-d, which is released due to execution of the machining permission command MPC-d, is inserted into the previous stage of the top block "N10 d machining" of the subsequent machining program NP2.

[0030] The comparison reference of the subsequent machining program NP2 is then shifted to the next block "N20 g machining", and the block to be compared of the preceding machining program NP1 is selected and shifted in order of the first block "N10 d machining", the second block "N20 g machining", the third block "N30 c machining", and the fourth block "N40 f machining". In this process, as shown in Fig. 5, a block "N60 e machining", which is determined to have no interference, is extracted, and a machining permission command MPC-g for permitting execution of the block "N20 g machining" of the subsequent machining program NP2 is inserted into the previous stage of the block "N60 e

machining". Furthermore, a machining permission wait command PWC-g, which is released due to execution of the machining permission command MPC-g, is inserted into the previous stage of the block "N20 g machining" of the subsequent machining program NP2. Thus, by repeating the above processing, in an example of Fig. 9, machining permission commands MPC-c and MPC-f are respectively inserted into the previous stages of the blocks N70 and N80 of the preceding machining program NP1, and machining permission wait commands PWC-c and PWC-f are respectively inserted into the previous stages of the blocks N30 and N40 of the subsequent machining program NP2.

[0031] The machining permission commands MPC-d to MPC-f can be instructed by various types of signs. For example, these commands are input by codes of "\$1 to \$4", which is a combination of dollar sign and an identification figure. Furthermore, the machining permission wait commands PWC-d to PWC-f can be also instructed by various types of signs. For example, these commands are input by codes of "¥1 to ¥4", which is a combination of sign ¥ and an identification figure. In this manner, the first and the second NC programs NP1 and NP2 for individual operation created by assuming that the left and right machining units LM and RM are individually operated to perform machining are modified for the coordinated operation of the both machining units. As a result, programs NP1R and NP2R for coordinated operation are created.

[0032] After the coordination information adding processing, second debugging DBG2 is executed as a fourth stage, as shown in Fig. 4. In the debugging DBG2, the CNC unit 50 executes the first and the second modified NC programs NP1R and NP2R for the coordinated operation simultaneously in parallel, to remove bugs in the both NC programs NP1R and NP2R under the state that the left and right machining units LM and RM are cooperatively operated. Since bug removal has finished in the second stage with respect to the programs NP1 and NP2 for individual operation, bugs found in the fourth stage can be determined as being due to the coordination information adding processing for the coordinated operation. As a result, the second debugging DBG2 in the fourth stage can be executed easily. When the modified programs NP1R and NP2R are executed by the CNC unit 50, the unit releases the sign

¥ corresponding to the identification figure, that is, the machining permission wait command PWC every time the unit decodes the respective ¥ signs, and starts NC execution processing 62 with respect to the block in the next stage of the sign ¥.

5 [0033] The main feature of the method according to the present invention is that the method includes a process for creating the individual machining programs NP1 and NP2 for respective units for individually operating the both machining units, and adding the coordination information to the individual machining programs for the coordinated operation in the next stage, at the time of creating
10 the NC programs for operating the left and right machining units LM and RM cooperatively. In this case, respective NC programs NP1R and NP2R modified for the coordinated operation include the coordination information for the coordinated operation, but are maintained as separate programs. Accordingly, there is such an advantage that when one of the machining units (for example,
15 LM) has a failure, the program (NP2) for individual operation created for the other machining unit (RM) and maintained is used, or the information (PWC) added for the coordinated operation is removed from the program (NP2R) for the coordinated operation for this unit and the program is used, to perform machining in the individual operation by the other machining unit. The method
20 of the present invention has an additional feature in that after the machining programs NP1 and NP2 for individual operation are created, debugging processing DBG1 is executed for these programs, and after the machining programs NP1R and NP2R for coordinated operation are created, debugging processing DBG2 is executed for these programs, thereby facilitating extraction
25 and removal of bugs.

[0034] The coordination information adding processing 64 can be manual processing performed by an operator or a programmer by operating an input/output unit 54. However, the present invention has such an additional feature that software for making the CNC unit 50 execute the adding processing
30 is incorporated in a system program STM of the CPU 51. Since the CNC unit 50 executes the adding processing 64 to realize automatization, the burden on the operator in creating the NC programs is reduced to the same level as that

when the machining units LM and RM are individually operated. As a result, remarkable technical progress can be realized such that the difficulty and induced program errors accompanying the conventional program creation method at the time of executing coordinated operation of the two machining units LM and RM can be eliminated.

[0035] Fig. 7 is a flowchart of interference preventing processing constituting a part of the coordination information adding processing 64 shown in Fig. 3.

When the operator inputs an execution command of the processing by operating the input/output unit 54, in the state that the NC programs NP1 and NP2 for individual operation are stored in the memory 52, the CPU 51 starts this processing. At first, at step S1, it is determined which one of the left and right machining units LM and RM is to execute the machining operation precedingly. This determination is performed by checking an X-axis target value of respective data blocks after the second block of one or both of the NC programs NP1 and NP2. Specifically, the CPU 51 determines a machining unit in which the X-axis target value shifts in a direction of returning to the own starting point as the preceding machining unit, and a machining unit in which the X-axis target value shifts in a direction going away from the own starting point as the subsequent machining unit. In the program example shown in Fig. 5, since the X-axis target values at a plurality of machining positions d, g, c, f, b, e, and a return to the starting point sequentially, the first NC program NP1 determines the left machining unit LM executing this as the preceding machining program NP1. Furthermore, since the X-axis target values at the machining positions d, g, c, f, b, e, and a move away from the starting point sequentially, the second NC program NP2 determines the right machining unit RM executing this as the subsequent machining program NP2.

[0036] When the left machining unit LM is determined to perform the preceding machining, memory areas ML and MR for storing the first and the second NC programs NP1 and NP2 respectively are set as an area B to be compared and a comparison reference area A (step S2). On the contrary, when the right machining unit RM is determined as the preceding machining, memory areas ML and MR are set as the comparison reference area A and the area B to be

compared (step S3). Addresses A_i and B_i of the both areas A and B are set to the first address of these areas (step S4). In Fig. 5, an initial value is expressed as address 0, but these addresses correspond to the first address of the areas A and B. Subsequently, the NC data blocks to be stored in the addresses A_i and B_i are read (steps S5 and S6), to determine interference between the left and right spindle heads 31 and 32 from the X-axis target value in these blocks (step S7). That is, it is assumed that the left and right spindle heads 31 and 32 are positioned simultaneously in these positions of the target value, and at this time, it is determined whether the spindle heads 31 and 32 physically come in contact with or collide with each other. This determination is specifically performed in the following manner. That is, absolute feed positions XL and XR of the spindle heads 31 and 32, using these starting points as a point of origin, is subtracted from a pre-measured distance XT between these left and right starting points, and when the difference is smaller than the predetermined interference threshold K, it is determined that there is an interference, and when the difference is larger than the predetermined interference threshold K, it is determined that there is no interference. The X-axis target value in the NC programs NP1 and NP2 for the left and right machining units is defined by assuming that the respective own starting points are 0 point, and designating a shift amount from the 0 point toward the other machining unit as a positive absolute amount.

[0037] When it is determined that there is an interference, the address B_i of the area to be compared is incremented by one address (step S8), and it is determined whether the data block at this new address is the last block (EOB) (step S9). When it is not the last block, the data block at the new address is read (step S6), and interference check is executed based on the read data block and the data block at the address A_i of the comparison reference area (step S7). Thus, the respective NC data blocks for the subsequent machining unit are designated as the comparison reference, and all NC data blocks for the preceding machining unit (blocks to be compared) are sequentially compared therewith.

[0038] In the program example in Fig. 5, when the right spindle head 32 performs tapping at position d, steps S6 to S9 are repetitively executed so as to check at which position of d, g, c, f, b, e, and a the left spindle head 31 that performs drilling interferes with the right spindle head 32. Consequently, when
5 it is determined that there is an interference, the address Bi of the area to be compared is sequentially incremented simply, to change the data block in the first NC program NP1 in order of N10, N20, N30, N40, When the data block at the address Bi of the area to be compared is changed to the one defining drilling at the position b, the relationship of $XT-SL-XR > K$ is realized, and it is
10 determined that there is no interference (step S7).

[0039] When it is determined that there is no interference, the first permission wait command ¥1 is inserted into the previous stage of the comparison reference address Ai, that is, the previous stage of the data block N10 for tapping at position d, and the first permission command \$1 is inserted into the
15 previous stage of the address Bi to be compared, that is, the previous stage of the data block N50 for drilling at position b (step S10). After this processing, the comparison reference address Ai is advanced by 1 (step S11), and it is determined whether the updated comparison reference address Ai is the last data block (step S12). When it is not the last block, the address Bi to be
20 compared is cleared to an initial value (step S13). Thereafter, control returns to step S5, to read the NC data block at the updated comparison reference address Ai, and comparison between the second data block for tapping of the second NC program NP2 for subsequent machining and the respective data blocks in the first NC program NP1 for preceding machining is executed as
25 described above.

[0040] By performing the processing as described above, in the machining example in Fig. 5, a second permission wait command ¥2 and a second permission command \$2 are respectively inserted into the previous stage of the data block defining the machining of tapping and drilling, so that simultaneous
30 machining of tapping at position g and drilling at position e are permitted. A third permission wait command ¥3 and a third permission command \$3 are respectively inserted into the previous stage of the data block defining tapping

at position c and drilling at position a, and further, a fourth permission wait command ¥4 and a fourth permission command \$4 are respectively inserted into the previous stage of the data block defining tapping at position f and an original position returning data block of the left machining unit LM that performs
5 preceding machining. In the flowchart of Fig. 7, when the block is determined to be the last block at step S9, control returns to step S5 after step S13. However, when the determination result is Yes at step S9, the control flow can be finished as shown by broken line, in order to omit further processing.

[0041] A program creation method when the left and right machining units LM

10 and RM coordinate with each other to machine the work W at respective indexed positions of the rotation table 41 is explained next, with reference to Figs. 8 to 10. Fig. 8 depicts the relationship between the first and the second individual machining NC programs NP1 and NP2 for the left and right machining units individually created and the indexed surfaces L1 to L3 of the work W. In
15 the respective NC programs NP1 and NP2, a table indexing command for directing one machined surface toward the spindle heads 31 and 32 is defined as an independent data block in the previous stage of a group of data blocks for machining the machined surface of the work W. Specifically, there are one data block indexing a first machined surface L1 of the work W to a machining
20 position facing the spindle heads 31 and 32, a group of data blocks for the first machined surface L1, one data block indexing a second machined surface L2 to a machining position, a group of data blocks for the second machined surface L2, one data block indexing a third machined surface L3 to a machining position, a group of data blocks for the third machined surface L3, and lastly, one data
25 block indexing the first machined surface L1 again to the machining position as carrying-out preparation of the work.

[0042] The data blocks in each group for machining the same machined surface of the work W correspond to the programs NP1 and NP2 explained with reference to Figs. 5 and 6. In this stage, however, the machining permission
30 command \$1, \$2, ... and the machining permission wait command ¥1, ¥2, ... paired therewith have not yet been inserted. Preferably, in a type using a known pallet changer to automatically change a work pallet on the rotation table

41, a pallet carrying-in command is defined as an independent data block in the forefront stage of respective NC programs NP1 and NP2, and a pallet carrying-out command is defined as an independent data block in the final stage thereof. [0043] In the respective NC programs NP1 and NP2 for individual machining
5 created in this manner, a command for indexing the rotation table 41 at the same position in the same order of the table indexing operation is defined in an overlapping manner. After the CNC unit 50 issues a command so as to index the rotation table 41 to one position according to one of the NC programs, if a drive circuit DUB is instructed to index the rotation table 41 to the same position
10 according to the other NC program, an error occurs in the indexing operation. Therefore, it is necessary to modify these individual machining NC programs NP1 and NP2 for coordinated operation, and the CNC unit 50 executes the coordination information adding processing 64 shown in Fig. 3.
[0044] Fig. 9 is an explanatory diagram of the outline of the whole coordination
15 information adding processing 64 executed by the CNC unit 50. Specifically, the coordination information adding processing 64 includes approximately three types of processing. In the first processing, the preceding machining unit and the subsequent machining unit are determined in the same processing as at step S1 in Fig. 7 based on the data block group for the same machined surface,
20 table indexing commands IA to IC in the previous stage of the data block group for one machining unit subjected to the subsequent machining and the last indexing command ID for returning the table are invalidated, and an indexing completion wait command IWC is inserted into the previous stage of the respective data block groups. In the program example in Fig. 9, the first and the
25 second NC programs NP1 and NP2 for the left and right machining units respectively includes a first to a third program units LP1 to LP3, RP1 to RP3, which execute machining operation in three same indexed positions. The program units LP1 to LP3 in the first NC program NP1 is programmed so as to allow the machining operation of the left machining unit LM to precede that of
30 the right machining unit RM according to the program units RP1 to RP3. Therefore, in this program example, all indexing commands IA, IB, IC, and ID overlapping on the indexing commands specified in the NC program NP1 for the

left machining unit LM are removed or invalidated. Furthermore, the indexing completion wait command IWC is inserted into the next stage of the respective invalidated indexing commands IA, IB, and IC, excluding the last indexing command ID. Preferably, in this processing, an indexing wait release command
5 MIC for releasing the corresponding indexing completion wait command IWC is inserted into the respective next stage of the remaining table indexing commands IA to IC.

[0045] The second processing in the coordination information adding processing
64 includes adding a machining completion wait command MWC instructing not
10 to execute the next indexing command of the rotation table 41, unless all machining operations by the left and right machining units LM and RM are completed on the same machined surface, and a machining wait release command MMC for releasing the machining completion wait command MWC.

[0046] Specifically, the machining completion wait command MWC for
15 confirming completion of machining at the indexed position by one of the NC programs is inserted into one block in the previous stage of respective indexing commands IB, IC, and ID for indexing the rotation table left in the other NC program to the next indexed position, and the machining wait release command MMC for releasing the corresponding machining completion wait command
20 MWC is inserted into the other NC program. The third processing is insertion processing of the machining permission wait command and the machining permission command between corresponding groups of NC data blocks for preventing interference between the left and right spindle heads 31 and 32, when the left and right machining units LM and RM cooperatively machine the
25 same machined surface. This processing is executed as described above with reference to Figs. 5 to 7.

[0047] The table indexing commands IA, IG, IC, and ID are left in the first NC program NP1 for the left machining unit and the table indexing operation is controlled by this program. In the second processing, therefore, the machining
30 completion wait command MWC for waiting for the completion of machining according to the program units RP1, RP2, and RP3 is inserted into the next stage of corresponding respective program units LP1, LP2, and LP3.

Furthermore, the machining wait release command MMC for releasing the machining completion wait command MWC is inserted into the next stage of corresponding respective program units RP1, RP2, and RP3. In the third processing, the machining permission wait command PWC and the machining permission command MPC for coordinated operation are added to the data block group of the corresponding to program unit in the manner described above.

[0048] When a part of the program units in the second NC program NP2, for example, RP2 is designated for the preceding operation, different from the program example in Fig. 9, the indexing command IB in the program unit RP2 is left in the first processing, and the indexing wait release command MIC is inserted into the next stage, as shown by chain line. On the other hand, the indexing command IB in the corresponding program unit LP2 of the NC program NP1 is removed or invalidated, and the indexing completion wait command IWC is inserted, as shown by chain line, into the previous stage of the data block group Lc following the indexing command. Furthermore, in the second processing, the machining permission wait command MWC is inserted into the next stage of the second program unit RP2, and the machining wait release command into the next stage of the second program unit LP2, respectively, as shown by chain line. As a result, the indexing command of the rotation table 41 can be provided only from one NC program NC1, as shown in Fig. 9, or the program unit that provides the indexing command can be changed for each program unit, so that the work load of the machining unit that performs preceding machining, that is, the work load shared by the left and right machining units LM and RM can be made substantially equal.

[0049] Fig. 10 is a flowchart for allowing the CPU 51 in the CNC unit 50 to execute the coordination information adding processing 64 including the above-described first to third processing. When execution of the processing 64 is instructed by the operator, the first and the second NC programs NP1 and NP2 for individual machining are divided in a unit of indexing command (step S21). Specifically, as shown by broken line in Fig. 9, in each program, one indexing command and the one or more program units following this are divided as one

group, and a top memory address for storing the NC data in each group is stored. A division number D_n is then set (step S22), and in the program example in Fig. 9, $D_n=3$ is set.

[0050] Subsequently, a processing group pointer P_n is set to an initial value "1" (step S23), to specify LP1 and LP2 as the first program units in the subsequent processing. Thereafter, it is determined whether the pointer P_n exceeds the division number D_n , that is, whether the processing has been completed for all groups (step S24), and when the processing has been completed, the processing based on the flowchart finishes. When the coordination information adding processing 64 has been not yet completed, it is determined whether the first and the second NC programs NP1 and NP2 both have the program unit at the first indexing position specified by the pointer P_n (step S25). When only one of the NC programs has the program unit, that is, it is determined that the machining operation only for one of the machining units is programmed at that table indexing position, control jumps to step S34, where 1 is added to the pointer P_n to specify the second program unit, and then returns to S24. Therefore, when it is programmed, at each indexed position of the table 41, so that only one machining unit individually performs machining, any modification is not carried out with respect to the program unit that specifies the indexing position.

[0051] However, when the program unit for allowing the left and right machining units LM and RM to perform machining operation at each indexed position is present in both NC programs NP1 and NP2, it is determined which machining unit performs preceding machining (step S26). This determination is performed in the same manner as the processing at step S1 shown in Fig. 7. When it is determined that the left machining unit LM performs preceding machining, the indexing command IA in the program unit RP1 for the right machining unit RM is deleted or invalidated (step 27). The indexing completion wait command IWC is then inserted into the next stage of the indexing command IA, and the indexing release command MIC is inserted into the next stage of the left indexing command IA (step S28). Furthermore, the machining completion wait command MWC for confirming the completion of machining operation according

to the program unit RP1 for the right machining unit RM is inserted into the end of the program unit LP1 for the left machining unit LM, and the machining wait release command MMC is inserted into the end of the program unit RP1 (step S29). On the contrary, when it is determined that the right machining unit RM performs preceding machining, the indexing command IA in the program unit LP1 for the left machining unit LM is deleted or invalidated (step 30). The indexing completion wait command IWC is then inserted into the next stage of the indexing command IA, and the indexing release command MIC is inserted into the next stage of the left indexing command IA in the program unit RP1 (step S31). Furthermore, the machining completion wait command MWC is inserted into the end of the program unit RP1, and the machining release command MMC is inserted into the end of the program unit LP1 (step S32). [0052] Subsequent to step S29 or S32, the interference preventing processing is executed based on the flowchart in Fig. 7 (step S33). As a result, when the program units LP1 and RP1 for the left and right machining units LM and RM are executed by the CNC unit 50, these program units LP1 and RP1 are modified so that the left and right machining units LM and RM are cooperatively operated, and a part of the end of the machining operation of the preceding unit and a part of the start of the machining operation of the subsequent machining unit are executed simultaneously in parallel, in the state that the both spindle heads 31 and 32 do not interfere with each other in the direction of X axis. When the interference preventing processing at step S33 is completed, the pointer Pn is advanced (step S34). As a result, the second program units LP2 and RP2 are specified as the next object to be modified, and processing at steps S25 to S33 is executed again for the second program units LP2 and RP2. After the coordination information adding processing with respect to the second program units LP2 and RP2, the pointer Pn specifies the third program units LP3 and RP3 (step S34). Therefore, when the processing at steps S25 to S33 is executed again, the third program units LP3 and RP3 are modified, and the coordination information relating to the indexing of the rotation table 41 is added as shown in Fig. 9.

(43)公開日 平成13年1月9日(2001.1.9)

テーマート* (参考)

3 0 1 J 5 H 2 6 9

3 0 3 Z

E

審査請求 未請求 請求項の数6 O L (全 16 頁)

(21)出願番号 特願平11-174218

(22)出願日 平成11年6月21日(1999.6.21)

(71)出願人 000003470
豊田工機株式会社
愛知県刈谷市朝日町1丁目1番地

(72)発明者 堀田 尊之
愛知県刈谷市朝日町1丁目1番地 豊田工
機株式会社内

(72)発明者 加藤 友也
愛知県刈谷市朝日町1丁目1番地 豊田工
機株式会社内

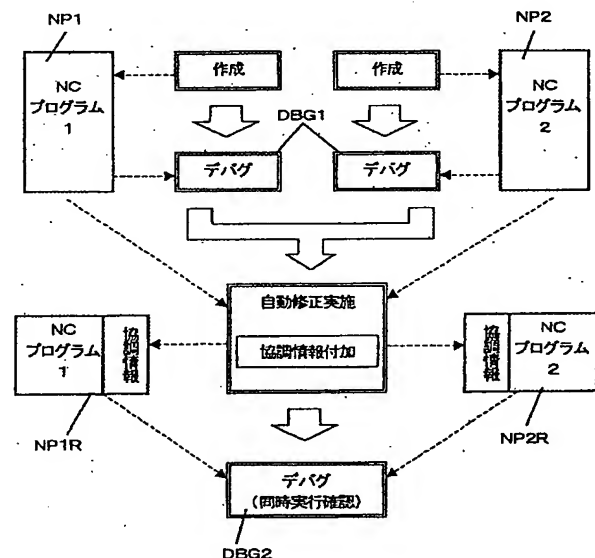
Fターム(参考) 5H269 AB01 AB03 AB06 CC02 CC13
CC15 DD01 EE01 EE07 KK10
QB06 QC01

(54)【発明の名称】 2主軸ヘッド工作機械用の加工プログラム作成方法

(57) 【要約】

【課題】 協調運転動作される左右加工ユニット用の第1及び第2加工プログラムの各々を互いに他の加工ユニットの動作を考慮せずに作成できるようにすること。

【解決手段】 第1ステップでは、左右の加工ユニットを単独運転するための第1及び第2NCプログラムNP1、NP2をそれぞれ個別に作成する。第2ステップでは、これらプログラムをCNC装置に個別に実行させ、単独運転における各プログラム中の不具合を抽出・修正する第1のデバッグ処理DBG1を行う。第3ステップでは、第1のデバッグを行った両プログラムを同時並行的に実行させるための協調情報をCNC装置により自動的に両プログラムに付加して協調運転用の第1及び第2NCプログラムNP1R、NP2Rを作成する。第4ステップでは、これら協調運転用の両プログラムをCNC装置に実行させて、協調運転における両プログラム中の不具合を抽出・修正する第2のデバッグ処理DBG2を行う。



【特許請求の範囲】

1
【請求項 1】 2 台の主軸ヘッドの各々を共通の移動範囲内で独立して移動すると共にこの移動方向を横切る方向に独立して移動し、前記共通移動範囲の中間位置で前記主軸ヘッドの前方に配置した単一のテーブル上のワークの複数の加工個所を前記 2 台の主軸ヘッドがそれらに取り付けた工具を用いて協調して加工するための 2 主軸ヘッド工作機械用加工プログラムの作成方法であって、加工動作を先行する一方の主軸ヘッドが前記複数の加工個所を他方の主軸ヘッドの起点側から自身の起点側に向かって復帰する途中で順次加工するための複数のデータブロックからなる先行加工プログラムを作成する第 1 加工プログラム作成ステップと、前記一方の主軸ヘッドの加工動作に後続する前記他方の主軸ヘッドが前記複数の加工個所又は他の複数の加工個所を自身の起点側から前記一方の主軸ヘッドの起点側に向かって移動する途中で順次加工するための複数のデータブロックからなる後続加工プログラムを作成する第 2 加工プログラム作成ステップと、前記一方の主軸ヘッドが実行する複数の加工個所に対する加工動作の一部と前記他方の主軸ヘッドが実行する前記複数の加工個所又は前記他の複数の加工個所に対する加工動作の一部とを並行動作させるように前記第 1 及び第 2 加工プログラム作成ステップで作成された前記先行及び後続加工プログラムの両方に協調運転のための協調情報を付加する協調情報付加ステップと、からなることを特徴とする加工プログラム作成方法。

【請求項 2】 請求項 1 に記載の方法において、前記協調情報付加ステップは前記工作機械を制御する CNC 装置が実行する自動処理ステップであることを特徴する加工プログラム作成方法。

【請求項 3】 請求項 1 又は 2 に記載の方法において、前記先行プログラムに従う前記一方の主軸ヘッドの移動と前記後続加工プログラムに従う前記他方の主軸ヘッドの移動とを個別に実行して前記先行及び後続加工プログラムの不具合を修正する単独加工デバッグステップを前記協調情報付加ステップに先立って実行し、また、前記協調情報付加ステップの後に、前記先行及び後続加工プログラムを並行して実行し前記 2 台の主軸ヘッドの移動を協調運転して前記先行及び後続加工プログラムの協調加工動作における不具合を修正する協調加工デバッグステップを実行することを特徴とする加工プログラム作成方法。

【請求項 4】 請求項 1～3 の何れか 1 項に記載の方法において、前記協調情報付加ステップは、前記後続加工プログラムの前半の一部の実行を待機させる加工許可待命令を前記後続加工プログラムの前半部に挿入すると共に前記加工許可待命令を解除する加工許可命令を前記先行加工プログラムの後半部に挿入することからなることを特徴とする加工プログラム作成方法。

【請求項 5】 2 台の主軸ヘッドの各々を共通の移動範

2
図内で独立して移動すると共にこの移動方向を横切る方向に独立して移動し、前記共通移動範囲の中間位置で前記主軸ヘッドの前方に配置した単一の回転テーブル上のワークの複数の加工個所を前記 2 台の主軸ヘッドがそれらに取り付けた工具を用いて協調して加工するための 2 主軸ヘッド工作機械用加工プログラムの作成方法であって、前記回転テーブルを複数の割出位置に順次割り出す複数の割出命令を定義すると共にこれら割出命令の各々に続いて前記主軸ヘッドの一方に前記ワーク上の複数の加工個所を加工させるための第 1 加工プログラムを定義する第 1 加工プログラム作成ステップと、前記回転テーブルを複数の割出位置に順次割り出す複数の割出命令を定義すると共にこれら割出命令の各々に続いて前記主軸ヘッドの他方に前記ワーク上の前記複数の加工個所又は他の複数の加工個所を加工させるための第 2 加工プログラムを定義する第 2 加工プログラム作成ステップと、前記第 1 及び第 2 加工プログラムに前記回転テーブルを同一割出位置に割り出す割出命令データが回転テーブルの同一割出順位に重複して存在するときは前記第 1 及び第 2 加工プログラム的一方に定義される前記重複する割出命令を無効にするプログラム修正ステップと、からなることを特徴とする加工プログラム作成方法。

【請求項 6】 請求項 5 に記載の方法において、前記プログラム修正ステップにおいて前記重複する割出命令が無効にされるときは協調情報をさらに付加し、この協調制御情報を付加する処理は、前記第 1 及び第 2 加工プログラム的一方に残された前記割出命令に基づいて前記回転テーブルが割出されたことを確認するために割出完了待命令を他方の加工プログラムの前記無効にされた割出命令に続く加工プログラムの前段に付加する操作と、前記無効化にされた割出命令に続く加工プログラムに基づく加工動作の完了を確認するために加工完了待命令を前記割出命令が残された加工プログラムの末尾に付加する操作と、前記加工完了待命令を解除する命令を前記無効化された割出命令に続く加工プログラムの末尾に付加する操作と、からなることを特徴する加工プログラム作成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、数値制御工作機械に使用される例えば NC プログラムで代表される加工プログラムの作成方法に関し、更に詳しくは、単一のワークに対し独立して運動できる 2 台の主軸ヘッドが協調して加工動作する加工プログラムの作成方法に関する。

【0002】

【従来の技術】 単一のワークに対し 2 台の主軸ヘッドが独立して移動して加工を施す公知の数値制御工作機械においては、2 台の主軸ヘッドの運動は別個の加工プログラムに従って制御される。従来、このような加工プログラムを作成する際、プログラマは、一方の主軸ヘッドの

運動制御のための第1加工プログラムを他方の主軸ヘッドの運動を勘案して作成し、また他方の主軸ヘッドの運動制御のための第2加工プログラムを前記一方の主軸ヘッドの運動を勘案して作成している。具体的には、第1加工プログラムを構成する複数のデータブロックの適切なブロック間及び第2加工プログラムを構成する複数のデータブロックの適切なブロック間に符合せ情報を入れ両主軸ヘッド同士が干渉せずに協調動作するようにしている。

【0003】

【発明が解決しようとする課題】しかしながら、従来のように一方の主軸ヘッドのための第1加工プログラムを他方の主軸ヘッドが行う動作を考慮して作成し、他方の主軸ヘッドのための第2加工プログラムを前記一方の主軸ヘッドが前記第1加工プログラムに従って行う動作を考慮して作成する方法では、プログラマーに過度の負担を課す面倒なプログラム作業となり、プログラムエラーを誘発し易い。特に、プログラム作成後に実際に工作機械を運動させて行うデバッグ作業では、第1及び第2プログラムを同時に実行させて行う必要があるため、同時並

行動作する2つの主軸ヘッドの運動を正確に観察すること自体が容易でなく、このため両プログラムに含まれるバグの特定が困難となる。

【0004】従って、本発明の主たる目的は、一方の主軸ヘッド用の第1加工プログラムを他方の主軸ヘッドが行う運動を考慮せずに作成でき、また他方の主軸ヘッド用の第2加工プログラムを前記一方の主軸ヘッドが行う運動を考慮せずに作成できるようにすると共に、かつプログラム作成後のデバッグ作業を容易にすることにある。本発明の他の目的は、両主軸ヘッドを協調運転する加工

【0005】

【課題を解決するための手段】上述した課題を解決するために、請求項1に記載の加工プログラム作成方法は、2台の主軸ヘッドが移動する共通の移動範囲の中間位置で前記主軸ヘッドの前方に配置した単一のテーブル上のワークの複数の加工個所を前記2台の主軸ヘッドがそれぞれに取り付けた工具を用いて協調して加工するための2主軸ヘッド工作機械用加工プログラムの作成方法を提供する。この方法は、先行する一方の主軸ヘッドが前記複数の加工個所を他方の主軸ヘッドの起点側から自身の起点側に向かって復帰する途中で順次加工するための複数のデータブロックからなる先行加工プログラムと、後続する他方の主軸ヘッドが前記複数の加工個所又は別の複数の加工個所を自身の起点側から前記一方の主軸ヘッドの起点側に向かって移動する途中で順次加工するための複数のデータブロックからなる後続加工プログラムとを個別に作成する。その後、前記一方の主軸ヘッドが実行する複数の加工個所に対する加工動作の一部と前記他方

の主軸ヘッドが実行する複数の加工個所に対する加工動作の一部とを並行動作させるように前記先行及び後続加工プログラムの両方に協調運転のための協調情報を付加することを特徴とする。

【0006】この加工プログラム作成方法では、最初に一方の主軸ヘッドが実行する先行加工プログラムと他方の主軸ヘッドが実行する後続加工プログラムとを、恰もこれら主軸ヘッドの各々を独立して加工動作するような形で互いに他の主軸ヘッドの動作を考慮せずに作成し、その後、先行加工プログラムの後半の一部と後続加工プログラムの前半の一部とを並行動作させるように両加工プログラムに協調運転のための協調情報を付加するようにした。

【0007】好適な形態として、請求項2に記載される発明は、前記協調運転のための協調情報の付加処理を、工作機械を制御するCNC装置に実行させることを特徴とする。これにより、加工プログラムの作成者への負担を単独運転用のプログラムの作成のみに軽減し、プログラムエラーの発生する機会を減少する。

【0008】さらに好適な形態として、請求項3に記載される方法は、単独運転用の加工プログラムを作成した後にこれら加工プログラムに基づいて両主軸ヘッドを個別に運転して各単独運転用加工プログラムに含まれるバグを除去し、このバグ除去を行った単独運転用加工プログラムに協調運転のための協調情報を付加すると共に、この付加処理の後に両主軸ヘッドをこの協調加工プログラムに基づいて協調運転させてこの協調運転における協調加工プログラム上のバグを除去することを特徴とする。デバッグ作業を単独運転用プログラムと協調加工プログラムと分けて2段階で行い、バグ除去を容易にしかつ短時間でできるようにする。

【0009】さらに好適な形態として請求項4に記載される発明は、前記協調情報を付加する処理では、前記後続加工プログラムの前半部の実行を待機させる加工許可待命令を前記後続加工プログラムの前半部に挿入すると共に、前記加工許可待命令を解除する加工許可命令を前記先行加工プログラムの後半部に挿入し、両主軸ヘッドの加工動作を一部並行動作させて加工能率の向上を図るようにした。

【0010】請求項5に記載の発明は、2台の主軸ヘッドが共通経路で移動する2主軸ヘッド工作機械においてワークを取り付けるテーブルを単一の回転テーブルとした場合に適用される加工プログラム作成方法が提供される。この場合、一方の主軸ヘッドのための単独加工プログラムを作成する際に、回転テーブルを複数の割出位置へ順次割り出す複数の割出命令を定義すると共に、これら割出命令の各々に続いて第1加工プログラムを定義し、また他方の主軸ヘッドのための単独加工プログラムを作成する際に、各割出命令の定義に続いて第2加工プログラムを定義する。さらに、協調運転のためのプログ

ラム修正においては、前記回転テーブルの割出順序の同一順位に同一割出位置を指定する割出命令が両加工プログラムに重複して存在するか否か判定し、重複する場合には両加工プログラムの一方向の重複する割出命令を無効にする。

【0011】これにより、2つの加工プログラムが同時期に同一割出位置へ回転テーブルを割り出すような制御上の干渉を排除する。ここで、割出命令を無効にするとは、その割出命令を削除したり、或いはCNC装置がその割出命令を実行せずに無視するように特別な記号を付加する等により実施できる。好適には、前記第1加工プログラム及び第2加工プログラムは、請求項1に定義されるような先行加工プログラム及び後続加工プログラムとし、またプログラム修正処理では請求項1に定義される協調情報の付加処理を行うようにしてもよい。

【0012】好適には、請求項6に記載されるように、重複する割出命令を無効にする場合、両加工プログラムに協調情報を付加する。具体的には、例えば、第1加工プログラムの無効にされなかった割出命令に基づいて回転テーブルの割出完了後に第2加工プログラムの無効にされた割出命令に続く加工プログラムが開始されるようにするため、この加工プログラムの前段に割出完了待命令を挿入し、また、この加工プログラムに基づく加工動作の完了を確認するために加工完了待命令を第1加工プログラムの無効にされなかった割出命令に続く加工プログラムの次段、つまりこの加工プログラムに続く次の割出命令の前段に挿入し、さらに、前記加工完了待命令を解除する命令を無効化された割出命令に続く加工プログラムの末尾に挿入する。

【0013】これにより、同一割出順位において回転テーブルが同一割出位置に割り出された状態では、2つの主軸ヘッドの加工動作がそれぞれの加工プログラムに従って異なる時点で開始され異なる時点で終了されても、それら加工プログラムに基づく2台の主軸ヘッドの加工動作が完了するまで何れか一方の加工動作が先に完了しても回転テーブルをその割出位置に維持し、回転テーブルの誤動作を排除する。前記割出完了待命令、加工完了待命令及びこの加工完了待命令を解除する命令を例えば”M機能”として周知の補助機能命令を用いれば、CNC装置に待合わせ制御のための特別なソフトウェアを付加せずに回転テーブルの割出しに関連して両主軸ヘッドを協調制御できる。前記割出完了待命令と加工完了待命令は、それぞれ異なる加工プログラムに挿入することが必要であるが、第1及び第2加工プログラムのどちらに割出完了待命令を付加するようにしてもよい。前記割出完了待命令は割出命令が無効にされる加工プログラムに挿入するが、この割出完了待命令を解除する命令を割出命令が有効に残される加工プログラムに挿入するようにしてもよい。この解除命令を使用しない形態では、回転テーブルが無効にされた割出命令により指定される回

転位置に割り出されたか否かを確認した後に前記割出完了待命令を解除するソフトウェアを設けることが好ましい。

【0014】

【発明の実施の形態】以下、本発明の実施の形態を図面に基いて説明する。図1は、本発明方法を実施する数値制御装置の制御対象である2台の主軸ヘッド付き工作機械10の正面図を示す。この図において、11は工作機械10のベッドであり、このベッド11の後部上面にガイドウェイ11Gが形成され、これに沿って左右加工ユニットLM、RMを構成する一対のコラム12、13が図において左右方向(X方向)に案内されている。コラム12、13は、それぞれサーボモータ14、15、送りネジ16、17、ナット18、19からなる左右送り機構により左右移動され、ベッド11の片端に設定された自身の送り起点から互いに相手のコラムに向かって前進移動すると共に、逆に相手のコラムから離間して自身の送り起点に向かって後退移動する。

【0015】サドル21、22は、コラム12、13の内側面で上下に延びるガイドウェイ12G、13Gに沿って上下方向(Y方向)に移動でき、それぞれサーボモータ23、24、送りネジ25、26、ナット27、28からなる上下送り機構により上下移動される。主軸ヘッド31、32は、サドル21、22の互いに対向する側面で図1の紙面垂直方向の水平に延びる前後ガイドウェイ21G、22Gに沿って前後方向(Z方向)に移動でき、それぞれ図略のサーボモータ、送りネジ35、36、図略のナットからなる前後送り機構により前後移動される。主軸ヘッド31、32は、前後移動方向と平行に延びるそれぞれの軸線の周りに工具主軸39、40を回転自在に支持し、それぞれに内蔵したビルトインモータにより工具主軸39、40を回転駆動できる。工具主軸39、40は、コラム12、13がそれぞれの起点に復帰した原位置状態において、図略の自動工具交換装置により、それらの前端に複数の工具T(図8参照)を選択的に装着できる。

【0016】回転テーブル41は、工具主軸39、40の前端側のベッド11上で、かつこのベッド11の左右幅方向の中間位置に設置され、図略の割出機構により水平面内で回転割出しされ、その上面に適宜治具を介して取り付けられる単一の工作物Wの各加工面を工具主軸39、40に向けることができる。従って、主軸ヘッド31、32は、それら間に定義される左右方向の共通の移動経路内でコラム12、13と共に左右移動され、またサドル21、22と共に上下動されて、工具主軸39、40をワークWの加工位置へ位置決めし、それら自身が前後移動して工具主軸39、40前端の図略の工具により、ワークWに加工を施すように動作する。ワークWに対する加工は、工具がドリルの場合では穴明け加工であり、タップの場合では加工済の穴内面にネジ切りを行

う。エンドミルの場合では工具とワークWとの当接位置で主軸ヘッド31、33を単独又は協調して互いに直交するX、Y及びZ方向に同時2軸又は3軸制御してワークWにコンタリング加工を施すことができる。必要であれば、回転テーブル41上に複数のワークを取付けるようにしてもよい。

【0017】図2は、上記工作機械10を制御する制御システムを示す。CNC装置50は、主として中央処理装置（以下、CPUと称する。）51、メモリー52からなり、メモリー52のROMに記憶されたシステムプログラムSTMの制御下で各種演算処理を実行する。メモリー52のRAMには、第1及び第2数値制御プログラム（以下、NCプログラム）NP1、NP2を記憶する領域、各種信号のオン・オフを記憶するフラグ領域FLG、後述する各種パラメータやデータK、XT、・・・などを記憶するデータ領域および演算処理を実行するワーキング領域WORKが設けられている。CPU51は、インタフェース53を介して入力装置54及びシーケンス制御用のプログラマブル・ロジック・コントローラ（以下、PLCと称する。）55に接続され、これらとの間で種々の情報を伝達或いは交換する。入力装置54は、各種の命令スイッチ類541、データ入力用のテンキー542、表示装置543等からなる。

【0018】また、CPU51は、インタフェース55を介して駆動装置56に接続される。この駆動装置56は、左加工ユニットLMのための駆動回路DUX、DUY、DUZ、右加工ユニットRMのための駆動回路DUU、DUV、DUW、さらには回転テーブル41の割出しのための駆動回路DUBから構成され、駆動回路DUX、DUY、DUZに接続されたサーボモータ14、23、33、駆動回路DUU、DUV、DUWに接続された15、24、34及び駆動回路DUBに接続されたサーボモータ42を制御し、これらモータの回転位置を検出するエンコーダ141、231、331、151、241、341、421から位置帰還出力を受取る。左加工ユニットLMのための駆動回路と右加工ユニットRMのための駆動回路とは、DUXとDUUが、DUYとDUVが、DUZとDUWがそれぞれ対応し、これらは左右送り制御軸、上下送り制御軸、前後送り制御軸を構成している。

【0019】CPU51が実行する処理機能の説明図である図3を参照して説明すると、CPU51は、入力されるNCプログラム及び各種パラメータをオペレータが対話形式で入出力装置54からメモリー52の各ユニット用の記憶エリアに記憶できるように支援する入出力支援処理61と、入力されたNCプログラムを1データブロックづつ読取・解読して目標位置、送り速度、制御モード等の設定や目標位置までの補間点の演算を行うNC実行処理62と、設定された目標位置、送り速度、制御モード等の情報及び補間点情報に基づき各制御軸が単位

時間毎に到達すべき位置決め点を演算し各制御軸に出力する移動命令処理63と、を左右の加工ユニットLM、RMの各々について、システムプログラムSTMの制御下で実行する。入出力支援処理61は、CNC装置51内で計算或いは記憶される各種情報を各加工ユニットLM、RM毎に区別して表示画面543に表示させる表示処理機能も実行する。NC実行処理62は、NCプログラムの各データブロックに含まれる工具主軸の回転起動・停止命令、クーラントのオン・オフ命令、工具交換命令等のM機能命令（補助機能命令）に関する情報をPLC55との間で所定時間毎に交換する処理機能も含む。

【0020】CPU51が実行するこれらの処理機能はこの種の工作機械を制御するCNC装置が持つ一般的機能として公知であるので、本発明に係る処理について以下に簡単に説明する。CPU51の処理時間は、例えば10ミリ秒のような単位時間毎に左加工ユニット用処理時間帯と右加工ユニット用処理時間帯とが時間的位相差を持って交互に割当てられるように時分割されている。CPU51は、左加工ユニット用処理時間帯の各々では、左加工ユニットLMのための第1NCプログラムNP1についてNC実行処理62又は移動命令処理63を実行し、右加工ユニット用処理時間帯の各々では、右加工ユニットRMのための第2NCプログラムNP2についてNC実行処理62又は移動命令処理63を実行する。

【0021】NC制御動作の開始直後及び1データブロックの制御動作完了直後の処理時間帯では、CPU51はNC実行処理62を行い、この処理において、NCプログラムNP1、NP2をメモリー52から1データブロック単位で読出・解読し、Gコード（制御コード）命令、制御軸、目標位置、送り速度等を設定し、輪郭制御の場合は目標位置までの補間点を演算し、また読出された1データブロック中にM機能命令が含まれる場合はPLC55とシーケンス制御情報を交換する。CPU51は、NC実行処理を実行した処理時間帯の単位時間後の次の同一加工ユニットに割当された処理時間帯では、移動命令処理63を実行し、設定された各種情報に基づき前述したコラム12、13、サドル21、22、主軸ヘッド31、32、回転テーブル41などの可動要素がそのブロックで指定される目標位置に指定された速度で移動されるように駆動装置56を制御する。

【0022】より具体的には、この移動命令処理63では、前記各種情報及び指定された制御軸のエンコーダからの帰還情報に基づきその指定された制御軸がこの割当時間内に到達すべき通過点を演算し、その駆動回路へ出力する。単位時間経過後に次の同一加工ユニットに割当された処理時間帯が始まる時、再び移動命令処理63を実行し、この移動命令処理63は同一加工ユニットの1データブロックで指定された制御軸が目標位置に到達し

てその1データブロックが完了するまで、繰り返し実行される。この1データブロックに基づくNC制御が完了すると、CPU 51は、この制御完了した加工ユニットに割り当てられた次の単位時間経過後の処理時間帯において、同一加工ユニットのためのNCプログラムの次の1ブロックデータについてNC実行処理62を行う。なお、入出力支援処理61は、各加工ユニットLM、RMの割当処理時間帯の後半部の空き時間を利用して表示画面543への表示処理を実行する。

【0023】本発明は、上述したように2台の主軸ヘッド31、32を協調動作させて単一のテーブル41上に取り付けられたワークWに加工を施す場合に両加工ユニットLM、RM用のNCプログラムを作成する方法であり、この方法を実施するために、CPU 51には図3のように協調情報付加処理64を遂行するためのソフトウェアが付加されている。本発明によるプログラム作成方法は、図4に処理上の特徴が示されており、CNC装置50の動作モードをプログラム作成モードに切り替えて以下の手順で実施される。

【0024】図4に示すように、先ず、第1段階として、左右の加工ユニットLM、RMが単独運転されてワークWを加工するための第1及び第2NCプログラムNP1、NP2を個別に作成する。両プログラムNP1、NP2の作成順序は、どちらを先に作成してもよい。図5は、下穴及びタップ加工のための両プログラムNP1、NP2の一例をワークWの加工個所a～gと関連して示す。この加工例では、左加工ユニットLM用のプログラムNP1は、シーケンス番号N10～N80までの8つのデータブロックで構成され、右加工ユニットRMに近い加工個所dからd→g→c→f・・・の順で自身の起点に復帰する過程で順次下穴を明けるようにプログラムされる。一方、右加工ユニットRM用のプログラムNP2は、同様にシーケンス番号N10～N80までの8つのデータブロックで構成され、相手側つまり左加工ユニットLM側に向かう過程で、前記下穴の加工順序と同一順序でこれら下穴にタップ加工するようにプログラムされる。

【0025】両プログラムNP1及びNP2は、前述した入出力支援処理61の実行の下で、例えば、表示画面543に指示される入力項目をテンキー542により入力することによりオペレータがCPU 51と対話する形式で作成される。このように作成される各NCプログラムNP1、NP2は、周知のEIAコードフォーマットで1データブロックづつメモリー52に記憶される。例えば、各データブロックには、シーケンス番号に続き、制御命令（所謂Gコード命令）、軸指定データ、目標位置データ、速度命令、M機能命令、ブロック終了命令の順で命令及びデータが入力される。各NCプログラムは、EIAコード以外のコードフォーマットを用いて作成してもよい。

【0026】図6は、左右加工ユニットLM、RMが複数の穴加工を分担して行う場合のプログラム例を示す。この場合、左加工ユニットLM用のプログラムNP1は、同図において上段に示す複数の一群の加工個所に、右加工ユニットRMに近い加工個所dからd→c→b→aの順で自身の起点に復帰する過程で順次穴明けするようにプログラムされ、右加工ユニットLM用のプログラムNP2は、下段に示す複数の一群の加工個所に、相手側つまり左加工ユニットLM側に向かう過程で、穴明けするようにプログラムされる。なお、図5及び図6に示す各NCプログラムにおいて、最初のブロックには工具主軸の回転速度命令も与えられる。また、工具交換命令は、各NCプログラムの最初のブロックの前段に与えられるが、これら図5及び図6では省略されている。さらに、図5及び図6において、単独運転用のプログラムNP1、NP2が作成される段階では、加工許可命令MPC及び加工許可待命令PWCは未だ挿入されていない。

【0027】再び図4に示すように、本発明の加工プログラム作成方法は、第2段階として、各NCプログラムNP1、NP2のデバッグが実行される。このデバッグ処理DBG1は、各NCプログラムをCNC装置50により個別に実行した場合にそのNCプログラムNP1、NP2中に含まれる不具合（バグ）を除去することを目的とする。この処理は、実際に工作機械10を動作させずに各NCプログラムNP1、NP2に基づきワークWに対して移動する工具の軌跡を表示画面543上に表示させて動作確認するソフトウェアシュミレーションによっても行うことができるが、より確実にデバッグを行う場合は、各NCプログラムNP1、NP2を実際にCNC装置50により個別に実行させ、ワークWに対し工具を移動させて実施される。この場合、工具をワークWに対し接触させない空運転でもよいが、工具によりワークWを加工する試し加工するときは、試し加工されたワークWの寸法を測定してそのNCプログラムNP1、NP2の良否を判定できる。

【0028】各NCプログラムNP1、NP2を単独に実行するデバッグDBG1に続き、第3段階では、両加工ユニットLM、RMを協調して同時に運転するために両NCプログラムNP1、NP2に協調情報が付加される。この処理は、図5及び図6を参照して説明される。下穴加工とタップ加工を両加工ユニットLM、RMに分担させる図5の場合では、加工動作を先行する第1NCプログラムNP1（以下、時折、先行加工プログラムと称する）に複数の個々に識別される加工許可命令MPCを付加し、加工動作が遅れて開始される第2NCプログラムNP2（以下、時折、後続加工プログラムと称する）に複数の個々に識別される加工許可待命令PWCを付加する。加工許可命令MPC及び加工許可待命令PWCの付加は、後続加工プログラムNP2のデータブロックを比較基準としてプログラム順に順次選択し、この比

較基準ブロックを先行加工プログラムNP1の順次選択される被比較データブロックと比較し、この比較結果に基づいて行われる。

【0029】より具体的には、後続加工プログラムNP2の最先データブロック「N10 d加工」を比較基準ブロックとして選択し、また被比較ブロックとして先行加工プログラムNC1の最初のデータブロック「N10 d加工」、2番目のブロック「N20 g加工」、3番目のブロック「N30 c加工」、4番目のブロック「N40 f加工」の順で選択し、選択された両ブロックのX軸目標位置の間隔が干渉を回避する所定値K以下であれば干渉有りとして判断し、所定値K以上であれば干渉無しと判定する。この所定値Kは、両主軸ヘッド31、32の主軸39と主軸40との間の左右方向における距離がこの値になった状態でも両主軸ヘッド32、32の互いの対向側面間に若干の空間ができるような値として予め設定される。先行加工プログラムNP1の選択ブロックは、最先ブロック近辺のものであれば、干渉有りとして判定されるが、最終ブロック側にその選択ブロックを移行する過程において、干渉無しと判定されるブロックが抽出される。この判定が変わるとき、後続加工プログラムNP2の最先ブロックの実行を許可する加工許可命令MPC-dを干渉無しに判定が変わるブロック「N50 b加工」の前段に挿入し、また、後続加工プログラムNP2の最先ブロック「N10 d加工」の前段に加工許可命令MPC-dの実行により解除される加工許可待命令PWC-dを挿入する。

【0030】次に、後続加工プログラムNP2の比較基準を次段のブロック「N20 g加工」にシフトすると共に、再び先行加工プログラムNP1の被比較ブロックを最先ブロック「N10 d加工」、2番目のブロック「N20 g加工」、3番目のブロック「N30 c加工」、4番目のブロック「N40 f加工」の順で選択してシフトする。この過程において、図5に例示されるように、干渉無しと判定されるブロック「N60 e加工」が抽出され、のブロックの前段に後続加工プログラムNP2のブロック「N20 g加工」の実行を許可する加工許可命令MPC-gを挿入し、また後続加工プログラムNP2のブロック「N20 g加工」の前段にこの加工許可命令により解除される加工許可待命令PWC-gを挿入する。このようにして、上述した処理を繰り返すことにより、図9の例では、加工許可命令MPC-c及びMPC-fを先行加工プログラムNP1のブロックN70、N80の前段にそれぞれ挿入すると共に、加工許可待命令PWC-c及びPWC-fを後続加工プログラムNP2のブロックN30、N40の前段にそれぞれ挿入する。

【0031】加工許可命令MPC-d～MPC-fは、各種の記号で指示できるが、例えばドル記号と識別数字との組み合わせ「\$1～\$4」のコードで入力され、加工

許可待命令PWC-d～PWC-fも各種の記号で指示できるが、例えば円記号と識別数字との組み合わせ「¥1～¥4」のコードで入力される。このようにして左右の加工ユニットLM、RMを単独に運転して加工することを想定して作成された単独運転用の第1及び第2NCプログラムNP1、NP2が両加工ユニットの協調運転用に修正され、協調運転用プログラムNP1R、NP2Rが作成される。

【0032】この協調情報付加処理の後、図4に示すように、第4段階として第2のデバグDBG2が実行される。このデバグDBG2においては、上記のように修正された協調運転のための第1及び第2NCプログラムNP1R及びNP2Rを同時並行的にCNC装置50に実行させ、左右の加工ユニットLM、RMが協調運転される状態での両NCプログラムNP1R、NP2R中のバグを除去する。単独加工用のプログラムNP1、NP2について第2段階でバグの除去が完了しているため、この第4段階で発見されるバグは、協調運転のための協調情報付加処理に起因するものと判断でき、このためこの第4段階における第2のデバグDBG2は容易に行うことができる。なお、修正された加工プログラムNP1R、NP2RがCNC装置50により実行されるとき、同装置は各\$記号を解説する毎に識別数字が対応する¥記号、つまり加工許可待命令PWCを解除し、この¥記号の次段のブロックについて、前述したNC実行処理62を開始する。

【0033】本発明方法の主たる特徴は、上述したように、左右の加工ユニットLM、RMを協調運転するNCプログラムを作成する場合に、両加工ユニットを単独運転するための各々のユニット用の単独加工プログラムNP1、NP2を最初作成し、次の段階においてこれら単独加工プログラムを協調運転用に協調情報を付加するプロセスを含む。この場合、協調運転用に修正されたそれぞれのNCプログラムNP1R、NP2Rは、協調運転のための協調情報を含むが別個のプログラムとして維持されるので、一方の加工ユニット（例えばLM）が故障した場合には、他方の加工ユニット（RM）用に作成しそのまま保存される単独運転用のプログラム（NP2）を使用するか、或いはこのユニット用の協調運転用のプログラム（NP2R）から協調運転のために付加した情報（PWC）を除去した後にこのプログラムを使用して他方の加工ユニットにより単独運転で加工を行うことができる利点がある。本発明方法は、単独運転用加工プログラムNP1、NP2を作成した後にこれらプログラムのバグ除去処理DBG1を実行し、また協調運転用加工プログラムNP1R、NP2Rを作成した後にさらにバグ除去処理DBG2を実行するようにし、バグの抽出及び除去を容易にした点を付加的な特徴の1つとしている。

【0034】協調情報付加処理64は、オペレータ又は

プログラマが入出力装置54を操作して行うマニュアル処理でもよいが、本発明はこの付加処理をCNC装置50に実行させるソフトウェアをCPU51のシステムプログラムSTMに組み入れたことを他の付加的特徴としている。この付加処理64をCNC装置50に実行させて自動化を図ることにより、NCプログラム作成におけるオペレータへの負担は、単に各加工ユニットLM、RMを単独で運転する場合と同程度になり、2台の加工ユニットLM、RMを協調運転する場合の従来プログラム作成方法に付随する困難性、プログラムエラーの誘発を排除できると云った顕著な技術的進歩が実現される。

【0035】図7は、図3に示す協調情報付加処理64の一部を構成する干渉防止処理のフローチャートである。単独運転用のNCプログラムNP1、NP2がメモリ52に記憶されている状態で、オペレータが入出力装置54によりこの処理の実行命令を入力すると、CPU51がこの処理を開始する。まず、ステップS1で左右の加工ユニットLM、RMのどちらが先行して加工動作を実行するか否かを判定される。この判定は、NCプログラムNP1、NP2の何れか一方又は両方の2番目以降の各データブロックのX軸目標値をチェックすることによりなされる。具体的には、X軸目標値が自己の起点に復帰する方向に変遷する加工ユニットを先行加工ユニットと判定し、自己の起点から順次遠ざかる方向に変遷する加工ユニットを後続加工ユニットと判定する。図5のプログラム例では、第1NCプログラムNP1は複数の加工箇所d、g、c、f、b、e、aのX軸目標値が順次起点側に復帰するので、これを実行する左加工ユニットLMを先行加工ユニットと判定し、また第2NCプログラムNP2は複数の加工箇所d、g、c、f、b、e、aのX軸目標値が順次起点から離れるので、これを実行する右加工ユニットRMを後続加工ユニットと判定する。

【0036】左加工ユニットLMが先行加工と判定されると、第1及び第2NCプログラムNP1、NP2をそれぞれ記憶するメモリ領域ML、MRを被比較領域B及び比較基準領域Aとして設定する(ステップS2)。逆に、右加工ユニットRMが先行加工と判定されると、メモリ領域ML、MRを比較基準領域A及び被比較領域Bとして設定する(ステップS3)。次に、両領域A及びBのアドレスAi及びBiをそれら領域の最初のアドレスにセットする(ステップS4)。図では初期値を零番地として表わしているが、これらの番地はそれぞれ両領域A及びBの最初の番地に相当する。続いて、両アドレスAi及びBiに記憶されるNCデータブロックが読出され(ステップS5及びS6)、これらブロック中のX軸目標値から左右の主軸ヘッド31、32の干渉を判定する(ステップS7)。つまり、これら目標値の位置に左右の主軸ヘッド31、32が同時に位置決めされると仮定し、その場合に両主軸ヘッド31、32が物理的

に接触或いは衝突するか否かを判定する。この判定は、具体的には、両主軸ヘッド31、32がそれぞれ左右の起点間の予め測定済みの距離XTからこれら起点を原点とする絶対送り位置XL、XRを減算し、その差が所定の干渉限界値Kより小さいときは干渉有り、大きいときは干渉無しと判定する。なお、左右の加工ユニット用のNCプログラムNP1、NP2中のX軸目標値は、それぞれ自己の起点を零点としこの零点からの相手側へ向かう移動量を正の絶対量として定義される。

10 【0037】干渉ありの場合、被比較領域アドレスBiが1番地増加され(ステップS8)、この新番地のデータブロックが最終ブロック(EOB)であるか否かを判定される(ステップS9)。最終ブロックでないとき、その新番地のデータブロックが読出され(ステップS6)、これと比較基準領域アドレスAiのデータブロックとに基づいて干渉チェックが実行される(ステップS7)。このように、後続加工ユニット用の各NCデータブロックを比較基準とし、これを先行加工ユニット用の全てのNCデータブロック(被比較ブロック)と順次比較する。

20 【0038】図5のプログラム例では、右主軸ヘッド32がd位置でタップ加工を行う時、ドリル加工を行う左主軸ヘッド31がd、g、c、f、b、e、aのどの位置で干渉するかをチェックするようにステップS6～S9が繰り返し実行される。これにより、干渉有りと判断される場合は、単に被比較領域アドレスBiを順次増加させ、第1NCプログラムNP1のデータブロックをN10、N20、N30、N40・・・と変えて行く。そして、被比較領域アドレスBiのデータブロックがb位置でのドリル加工を定義するものになる時、XT-XL-XR>Kの関係となって干渉無しと判定される(ステップS7)。

30 【0039】干渉無しと判定されるとき、比較基準アドレスAiの前段、つまりd位置でのタップ加工用データブロックN10の前段に第1番目の許可待命令¥1が挿入され、被比較アドレスBiの前段、つまりb位置でのドリル加工用データブロックN50の前段に第1番目の許可命令\$1が挿入される(ステップS10)。この処理の後、比較基準アドレスAiが1番地進められ(ステップS11)、この更新された比較基準アドレスAiが最終データブロックか否かを判定される(ステップS12)。最終ブロックでないとき、被比較アドレスBiを初期値にクリアし(ステップS13)。その後ステップS5に復帰し、更新された比較基準アドレスAiのNCデータブロックを読出し、後続加工用の第2NCプログラムNP2の2番目のタップ加工用のデータブロックと先行加工用の第1NCプログラムNP1の各データブロックとの比較を上述したように実行する。

40 【0040】このように処理することにより、図5の加工例では、g位置のタップ加工とe位置のドリル加工の

同時加工を許容するように、これらの加工を定義するデータブロックの前段に第2許可待命令 $\$2$ と第2許可命令 $\$2$ がそれぞれ挿入され、c位置のタップ加工とa位置のドリル加工を定義するデータブロックの前段に第3許可待命令 $\$3$ と第3許可命令 $\$3$ がそれぞれ挿入され、さらにf位置でのタップ加工を定義するデータブロックと先行加工動作する左加工ユニットLMの原位置復帰データブロックの前段に第4許可待命令 $\$4$ と第4許可命令 $\$4$ がそれぞれ挿入される。尚、図7のフローチャートではステップS9で最終ブロックと判定されるときは、ステップS13の後にステップS5に復帰するようにしているが、ステップS9でYESと判定されるときは、それ以上の処理を省略するため破線で示すようにこの制御フローを終了するようにしてもよい。

【0041】次に、回転テーブル41の各割出位置において左右の加工ユニットLM、RMが協調してワークWを加工する場合のプログラム作成方法を図8～図10を参照して説明する。図8は、それぞれ個別に作成される左右の加工ユニット用の単独加工用の第1NCプログラムNP1と第2NCプログラムNP2とワークWの割出面L1～L3との関係を示す説明図である。各NCプログラムNP1、NP2は、ワークWの1つの加工面を加工する1群のデータブロックの前段にその加工面を主軸ヘッド31、32に向けるためのテーブル割出命令が独立したデータブロックとして定義されている。具体的には、ワークWの第1加工面L1を主軸ヘッド31、32と対面する加工位置へ割り出す1つのデータブロック、第1加工面L1のための1群のデータブロック、第2加工面L2を加工位置へ割り出す1つのデータブロック、第2加工面L2のための1群のデータブロック、第3加工面L3を加工位置へ割り出す1つのデータブロック、第3加工面L3のための1群のデータブロックと続き、そしてワークの搬出準備として最後に第1加工面L1を再び加工位置へ割り出す1つのデータブロックとで構成される。

【0042】ワークWの同一加工面を加工するための各1群のデータブロックは、例えば、図5及び図6を参照して説明したプログラムNP1、NP2に相当する。しかし、この段階では、図5及び図6に示す加工許可命令 $\$1$ 、 $\$2$ ・・・及びこれと対をなす加工許可待命令 $\$1$ 、 $\$2$ ・・・は未だ挿入されていない。好ましくは、公知のバレット交換装置を用いて回転テーブル41上にワークバレットを自動交換する形式のものとする場合は、各NCプログラムNP1、NP2の最前段にバレット搬入命令が独立したデータブロックとして定義され、最終段にバレット搬出命令が独立したデータブロックとして定義される。

【0043】このように作成される単独加工用の各NCプログラムNP1、NP2には、テーブル割出動作の同一順位に回転テーブル41を同一の位置に割り出す命令

が重複して定義されることになる。CNC装置50が一方のNCプログラムに従って回転テーブル41を1つの位置に割出すように命令を与えた後に、他方のNCプログラムに従って同一の位置へ割出すように駆動回路DUBが命令されると、割出動作上のエラーを生じる。このため、これら単独加工用のNCプログラムNP1及びNP2を協調動作のために修正する必要があり、CNC装置50は図3の協調情報付加処理64を実行する。

【0044】図9は、CNC装置50が実行する協調情報付加処理64全体の概要を説明する説明図である。具体的には、この付加処理64は、概略3つの処理からなる。第1の処理は、加工面が同一のデータブロック群に基づき先行加工ユニット及び後続加工ユニットを前述した図7のステップS1と同様な処理により判別し、後続加工する一方の加工ユニット用のデータブロック群の前段にあるテーブル割出命令IA～IC及びテーブル戻しのための最後の割出命令IDを無効にすると共に、各データブロック群の前段に割出完了待命令IWCを挿入する処理である。図9のプログラム例では、左右の加工ユニット用の第1及び第2NCプログラムNP1、NP2は、いずれも3つの同一の割出位置で加工動作を実行する第1～第3プログラム部LP1～LP3、RP1～RP3で構成され、第1NCプログラムNP1のプロプログラム部LP1～LP3は、左加工ユニットLMの加工動作をプログラム部RP1～RP3に従う右加工ユニットRMのそれらに対して先行させるようにプログラムされている。従って、このプログラム例では、左加工ユニットLMのNCプログラムNP1中に指定される割出命令と重複する割出命令IA、IB、IC、IDが全て除去されるか、無効にされる。また、最後の割出命令IDを除く無効にされた各割出命令IA、IB、ICの次段に割出完了待命令IWCが挿入される。好適には、この処理では、残したテーブル割出命令IA～ICの各々の次段に、対応する前記割出完了待命令IWCを解除する割出待解除命令MICを挿入する。

【0045】協調情報付加処理64の第2の処理は、同一加工面において左右の加工ユニットLM、RMの加工動作が全て完了しない限り回転テーブル41の次の割出命令を実行しないようにする加工完了待命令MWCとこれを解除する加工待解除命令MMCを追加することからなる。

【0046】具体的には、一方のNCプログラムに残される回転テーブルを次の割出位置に割出すための各割出命令IB、IC及びIDの前段の1ブロックに他方のNCプログラムによるその割出位置での加工完了を確認する加工完了待命令MWCを挿入し、対応する加工完了待命令MWCを解除する加工待解除命令MMCを他方のNCプログラムに挿入する。第3の処理は、同一加工面を左右の加工ユニットLM、RMが協調して加工動作する場合に左右の主軸ヘッド31、32同士の干渉を防止す

るための対応する 1 群の NC データブロック間での加工許可待命令と加工許可命令の挿入処理であり、この処理は図 5～図 7 を参照して上述したように実行される。

【0047】テーブル割出命令 I A、I B、I C 及び I D を左加工ユニット用の第 1 NC プログラム NP 1 に残してテーブル割出動作をこのプログラムにより制御するようにしたので、第 2 の処理では、プログラム部 RP 1、RP 2、RP 3 に従う加工の完了に待機する加工完了待命令 MWC を対応するプログラム部 LP 1、LP 2、LP 3 の各々の次段に挿入する。また、加工完了待命令 MWC を解除する加工待解除命令 MMC を対応するプログラム部 RP 1、RP 2、RP 3 の各々の次段に挿入する。第 3 の処理では、対応するプログラム部のデータブロック群に、協調運転のための加工許可待命令 PWC 及び加工許可命令 MPC を前述したように付加する。

【0048】図 9 のプログラム例とは異なって第 2 NC プログラム NP 2 の一部のプログラム部、例えば RP 2 を先行動作のためのものとする場合では、第 1 の処理において同プログラム部 RP 2 の割出命令 I B を残すと共に次段に鎖線で示すように割出待解除命令 MIC を挿入し、一方、第 1 NC プログラム NP 1 の対応するプログラム部 LP 2 の割出命令 I B を除去又は無効にすると共にこの割出命令に続くデータブロック群 Lc の前段に鎖線で示すように割出完了待命令 IWC を挿入する。さらに第 2 の処理において、加工完了待命令 MWC を第 2 プログラム部 RP 2 の次段に、加工待解除命令を第 2 プログラム部 LP 2 の次段に、それぞれ鎖線で示すように挿入する。これにより、回転テーブル 41 の割出命令は、図 9 に示すように一方の NC プログラム NC 1 のみから与えるようにすることもできるが、この割出命令を与えるプログラム部をプログラム部毎に変えるようにし、先行加工動作する加工ユニット、つまり左右の加工ユニット LM、RM が分担する作業を略均等にすることもできる。

【0049】図 10 は、前述した第 1～第 3 の処理を含む協調情報付加処理 64 を CNC 装置 50 の CPU 51 に実行させるためのフローチャートを示す。この処理 64 の実行がオペレータにより命令されると、先ず単独加工用の第 1 及び第 2 NC プログラム NP 1、NP 2 が割出命令単位で分割される（ステップ S 21）。具体的には、図 9 で破線で囲んで示すように、各プログラムは、1 つの割出命令とこれに続く 1 以上のプログラム部が 1 つのグループとして分割され、各グループの NC データを記憶する先頭のメモリーアドレスが記憶される。次に、分割数 Dn が設定され（ステップ S 22）、図 9 のプログラム例では、Dn = 3 と設定される。

【0050】続いて、処理グループポインタ Pn が初期値“1”にセットされ（ステップ S 23）、以降の処理の対象を第 1 番目のプログラム部として LP 1、RP 1 を指定する。この後、ポインタ Pn が分割数 Dn を超え

たか否か、つまり全てのグループについて処理が完了したか否か判定され（ステップ S 24）、完了のときこのフローチャートに基づく処理が終了する。協調情報付加処理 64 が未完了のとき、ポインタ Pn で特定される第 1 番目の割出位置について第 1 及び第 2 NC プログラム NP 1、NP 2 が共にプログラム部を有しているか判定され（ステップ S 25）、一方しかプログラム部がないとき、つまりそのテーブル割出位置では一方の加工ユニットのみの加工動作がプログラムされていると判定されるとき、ステップ S 34 へジャンプし、ここでポインタ Pn に 1 を加算して 2 番目のプログラム部を指定させた後、ステップ S 24 へ復帰する。このため、テーブル 41 の各割出位置において一方の加工ユニットのみが単独で加工動作するようにプログラムされる場合では、この割出位置を指定するプログラム部に対しては何らの修正も行われない。

【0051】しかしながら、各割出位置において左右の加工ユニット LM、RM が加工動作を行うようにするためプログラム部が両 NC プログラム NP 1、NP 2 に存在する場合は、どちらの加工ユニットが加工動作を先行するのかが判定される（ステップ S 26）。この判定は、図 7 のステップ S 1 における処理と同様に行われる。左加工ユニット LM を先行すると判定されるとき、右加工ユニット RM 用のプログラム部 RP 1 の割出命令 I A が削除又は無効にされ（ステップ S 27）、この割出命令 I A の次段に割出完了待命令 IWC が挿入されると共に、残された割出命令 I A の次段に割出解除命令 MIC が挿入される（ステップ S 28）。さらに、左加工ユニット LM 用のプログラム部 LP 1 の末尾に右加工ユニット RM の前記プログラム部 RP 1 に従う加工動作の完了を確認する加工完了待命令 MWC が挿入されると共に、プログラム部 RP 1 の末尾に加工待解除命令 MMC が挿入される（ステップ S 29）。逆に、右加工ユニット RM を先行すると判定されるとき、左加工ユニット LM 用のプログラム部 LP 1 の割出命令 I A が削除又は無効にされ（ステップ S 30）、この割出命令 I A の次段に割出完了待命令 IWC が挿入されると共に、プログラム部 RP 1 の残された割出命令 I A の次段に割出解除命令 MIC が挿入される（ステップ S 31）。さらにプログラム部 RP 1 の末尾に加工完了待命令 MWC が挿入されると共に、プログラム部 LP 1 の末尾に加工解除命令 MMC が挿入される（ステップ S 32）。

【0052】ステップ S 29 又は S 32 の後、前述した干渉防止処理が図 7 のフローチャートに基づいて実行され（ステップ S 33）、これにより左右の加工ユニット LM、RM 用のプログラム部 LP 1 と RP 1 は、これらが CNC 装置 50 により実行されるとき、左右の加工ユニット LM、RM が協調動作されて先行するユニットの加工動作の終側の一部と後続する加工ユニットの前側の一部とが両主軸ヘッド 31、32 が X 軸方向で干渉しな

いような状態で同時並行的に実行するように修正される。干渉防止処理 S 33 が完了すると、ポイント P_nは進められ（ステップ S 34）、これにより次の修正対象が 2 番目のプログラム部 L P 2、R P 2 に特定され、これらプログラム部 L P 2、R P 2 についてステップ S 25～S 33 が再度実行される。2 番目のプログラム部 L P 2、R P 3 に対する協調情報付加処理の後、ポイント P_nが 3 番目のプログラム部 L P 3、R P 3 を指定する（ステップ S 34）ので、再びステップ S 25～S 33 が実行されるとき、これら 3 番目のプログラム部 L P 3、R P 3 が修正され、回転テーブル 41 の割出に関する協調情報が図 9 のように付加される。

【0053】

【発明の効果】以上詳述したように、請求項 1 の発明によれば、先行する一方の主軸ヘッドが複数の加工個所を順次加工するための複数のデータブロックからなる先行加工プログラムと後続する他方の主軸ヘッドが前記複数の加工個所又は別の複数の加工個所を順次加工するための複数のデータブロックからなる後続加工プログラムとをそれぞれ個別に作成し、その後先行及び後続加工プログラムの両方に協調動作のための協調情報を付加するようにしたので、各主軸ヘッドが単独で実行すべき加工プログラムの作成は通常の単一主軸ヘッド工作機械における加工プログラムの作成と同様に比較的容易に作成でき、また協調運転のための協調情報の付加処理は両主軸ヘッドの干渉のみを考慮してこの干渉を回避するための協調情報を必要個所に挿入することにより達成されるので、プログラム作成上の誤りを容易に排除でき、全体としてプログラムの作成が容易となる。

【0054】好適には、請求項 2 に記載されるように、前記協調運転のための協調情報付加処理を、工作機械を制御する CNC 装置に実行させるようし、加工プログラム作成者への負担を単独運転用のプログラムの作成のみに軽減し、プログラムエラーの発生する機会が減少される。さらに好適には、請求項 3 に記載されるように、単独運転用の加工プログラムを作成した後にこれらプログラムに含まれるバグを除去し、このバグ除去を行った単独運転用加工プログラムに協調情報を付加して修正すると共にその後協調運転プログラム上のバグを除去するようにしたので、各運転時における不具合が単独運転用プログラム中に含まれるバグによるものか或いは協調運転プログラム中に含まれるバグによるものと云った観点でオペレータがバグを特定する作業が容易となり、全体としてバグの除去を容易かつ短時間でできるようになる。

【0055】さらに好適には、請求項 4 に記載されるように、前記協調情報は、後続加工プログラムの前半部の実行を待機させる加工許可待命令を後続加工プログラムの前半部に挿入すると共に、加工許可待命令を解除する加工許可命令を先行加工プログラムの後半部に挿入する

ようにしたので、両主軸ヘッドの加工動作の一部を両ヘッドが干渉しないような状態で並行動作でき、加工能率が向上される。請求項 5 の発明によれば、回転テーブルの割出順序の同一順位に同一割出位置を指定する割出命令が両加工プログラムに重複する場合には両加工プログラムの一方からその重複する割出命令を削除するようにしたので、2 つの加工プログラムが同時期に同一割出位置へ回転テーブルを割り出すような制御上の干渉が排除され、回転テーブルの割出制御における誤動作が排除される。

【0056】好適には、請求項 6 に記載されるように、重複する一方の割出命令の削除に関連して、割出命令が無効にされた加工プログラムの前段に割出完了待命令を挿入し、また、この加工プログラムに基づく加工動作の完了を確認するための加工完了待命令を割出命令が無効にされなかった加工プログラムの次段に挿入するようにしたので、2 つの主軸ヘッドの加工動作がそれぞれの加工プログラムに従って異なる時点で開始され異なる時点で終了されても、それら加工プログラムに基づく 2 台の主軸ヘッドの加工動作が完了するまでは何れか一方の加工動作が先に完了しても回転テーブルはその割出位置に維持され、回転テーブルの誤動作が排除される。

【図面の簡単な説明】

【図 1】本発明の加工プログラム作成方法を実施する 2 主軸ヘッド工作機械の概略を示す正面図。

【図 2】前記工作機械の制御システムを示すブロック線図。

【図 3】図 2 に示す CPU が実行する処理を説明するための機能ブロック図。

【図 4】本発明方法の概要手順を説明する説明図。

【図 5】本発明方法が適用されるプログラムの 1 例を示す説明図。

【図 6】本発明方法が適用されるプログラムの別の例を示す説明図。

【図 7】前記 CPU が実行する干渉防止処理ルーチンの詳細を示すフローチャート。

【図 8】回転テーブルを使用する場合のプログラムの 1 例を示す説明図。

【図 9】図 8 のプログラム例における協調情報付加処理を説明する説明図。

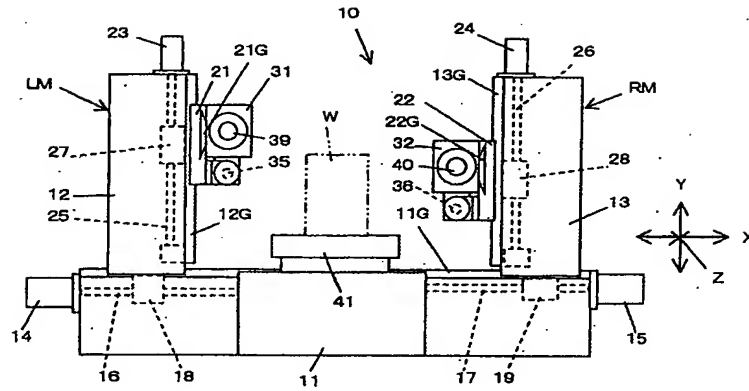
【図 10】回転テーブルを使用する場合において前記 CPU が実行する協調情報付加処理ルーチンの詳細を示すフローチャート。

【符号の説明】

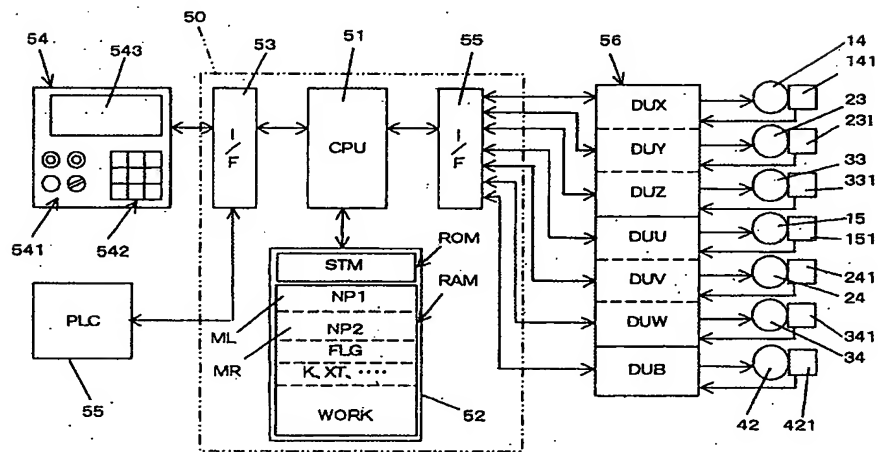
LM：左加工ユニット、 RM：右加工ユニット、 31、32：主軸ヘッド、 41：回転テーブル、 T：工具、 W：ワーク、 50：CNC 装置、 51：CPU、 52：メモリー、 56：駆動装置、 NP1、NP2：第 1 及び第 2 NC プログラム（先行及び後続加工 NC プログラム）、 DBG1、DBG2：第 1 及び

第2デバッグ処理、 64: 協調情報付加処理。

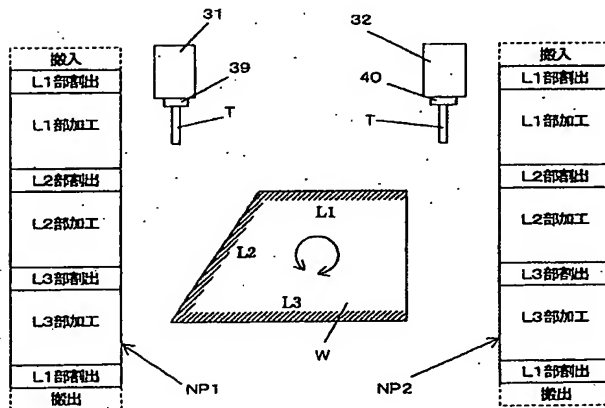
【図1】



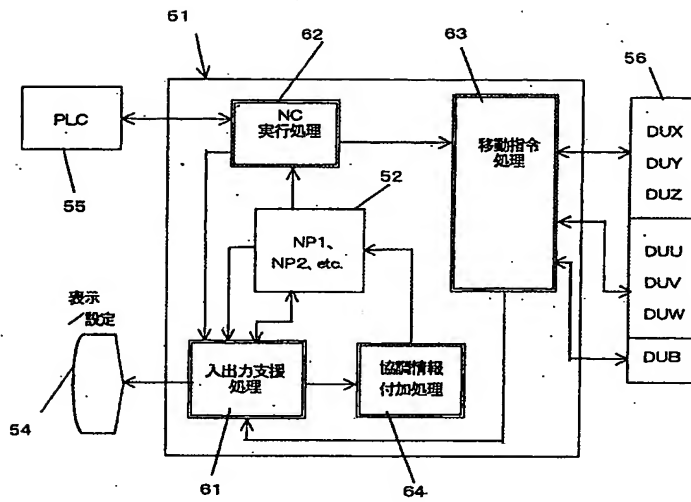
【図2】



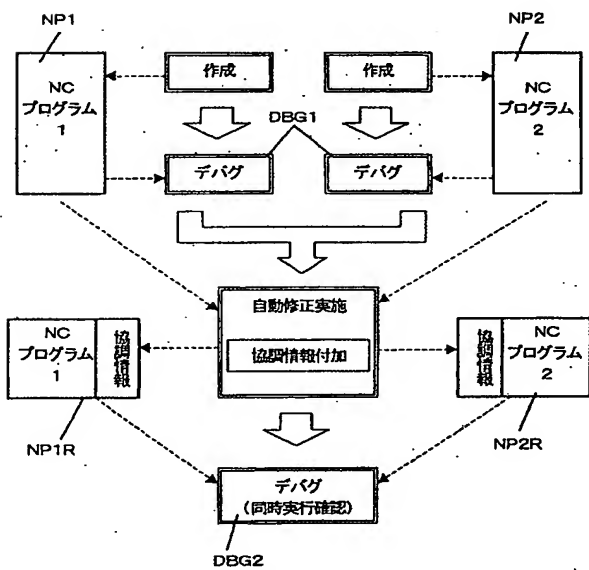
【図8】



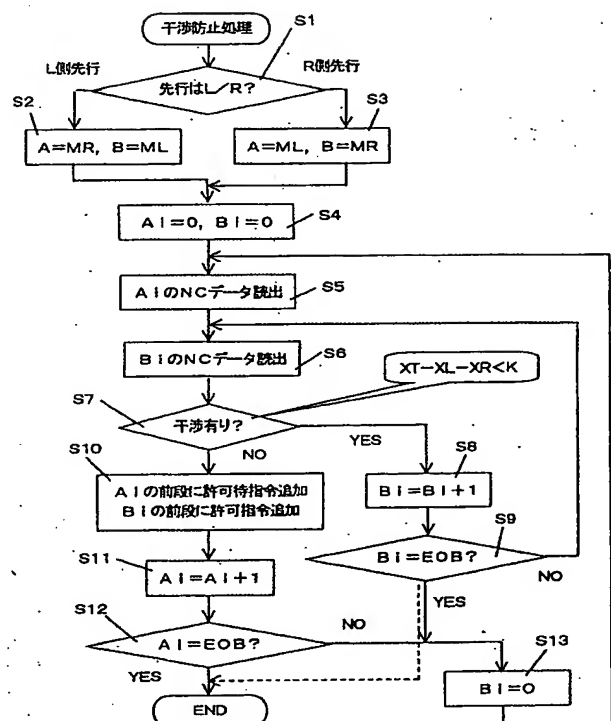
【図3】



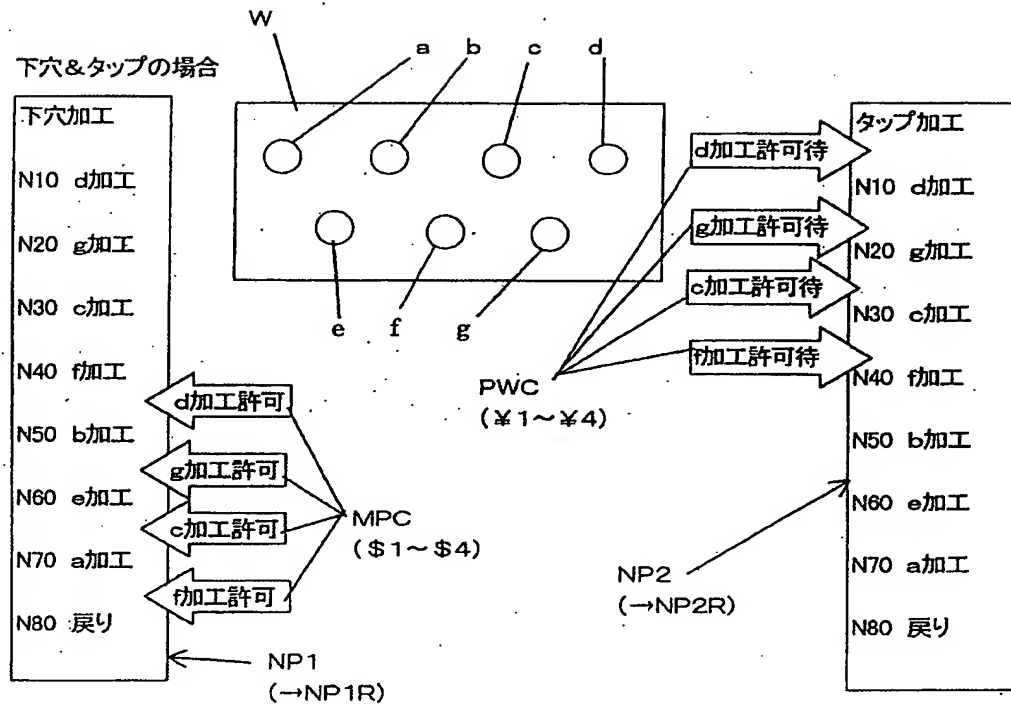
【図4】



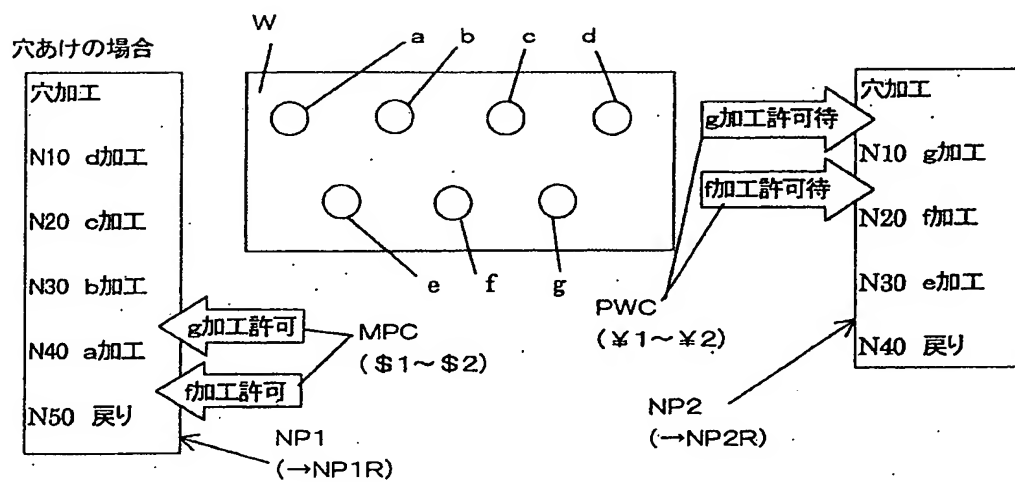
【図7】



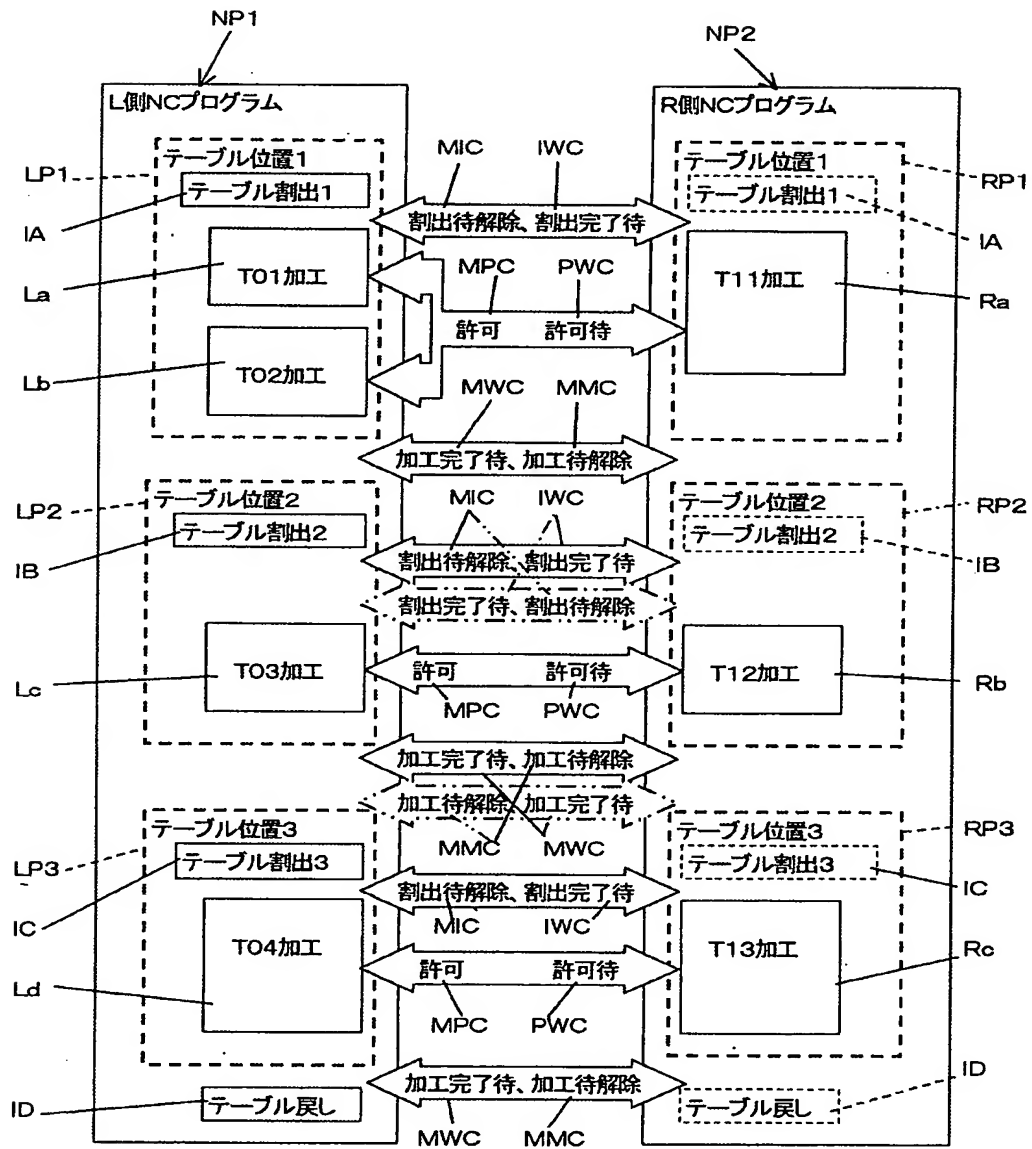
【図5】



【図6】



【図9】



【図10】

